CIND830
Python Programming for Data Science
Lab 8 - Questions

1 Arrays

a. The following class defines an array object with a fixed size of elements. However, executing each of the following commands either raises an error or generates an incorrect output. Extend the `iArray` class by defining the proper function(s) to resolve those raised errors.

```
class iArray():
    '''A representation of an array'''
    def __init__(self, capacity, fillValue = None):
        '''Capacity is the static size of the array.
        None object will be placed at each position.'''
        self._items = list()
        for count in range(capacity):
            self._items.append(fillValue)
```

i.
```
a = iArray(5)
len(a)
'''TypeError: object of type 'iArray' has no len()'''
```

ii.
```
a = iArray(5)
print(str(a).split())
'''Output: ['<__main__.iArray', 'object', 'at',\
   '0x7fd048283310>']'''
```

iii.
```
a = iArray(5)
a[3] = 2
'''TypeError: 'iArray' object does not support item assignment'''
```

iv.
```
a = iArray(5)
print(a[1])
'''TypeError: 'iArray' object is not subscriptable'''
```

v.
```
a = iArray(5)
for i in a:
    print(i)
'''TypeError: 'iArray' object is not iterable'''
```

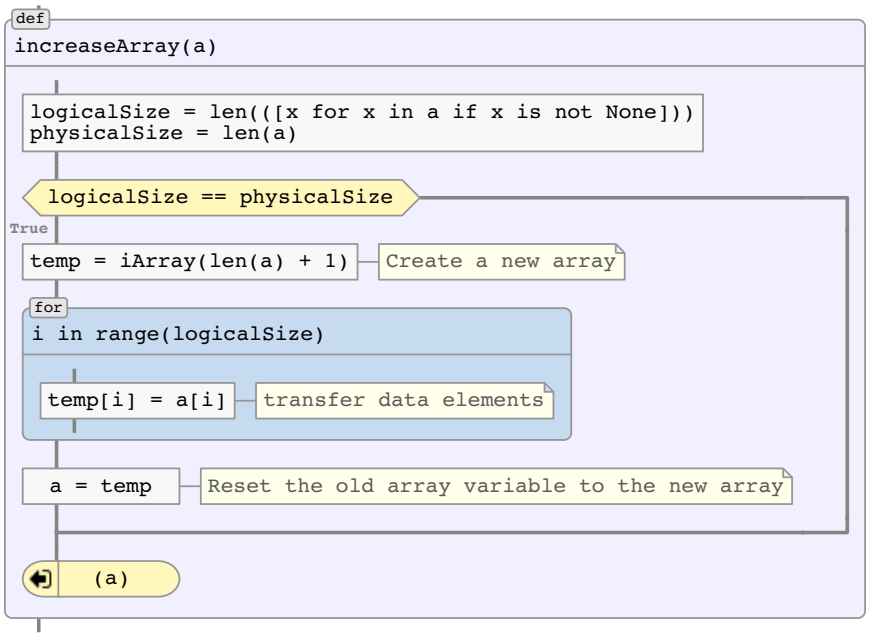
b. The following command gives an `IndexError` indicating that the assignment index is out of range.

```
a = iArray(5)
a[5] = 7
```

To resolve that error, the size of the array needs to be increased. Define a function that resizes a given array using the following steps:

- Check the physical and the logical sizes of the given array.
- Create a new larger array if needed.
- Copy the data from the old array to the new array.
- Reset the old array object to the new array object.

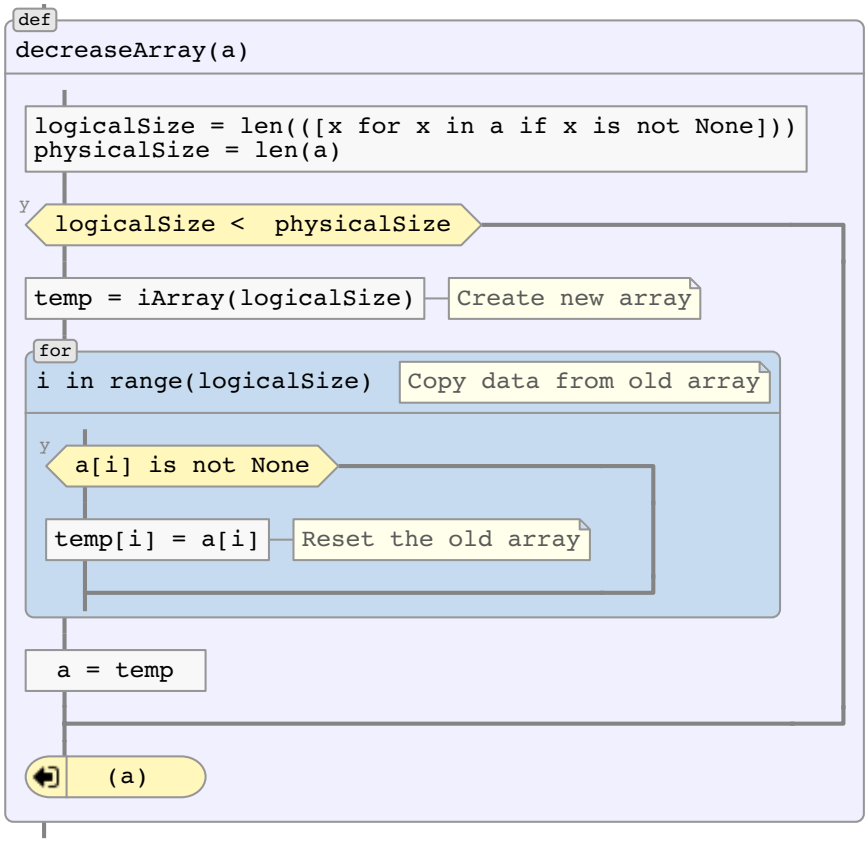
Hint: The following control-flow chart might be of help:



c. When the logical size of an array is less than the physical size, cells in memory go to waste. To resolve that error, the size of the array needs to be decreased. Define a function that resizes a given array using the following steps:

- Check the physical and the logical sizes of the given array.
- Create a new smaller array if needed.
- Copy the data from the old array to the new array.
- Reset the old array object to the new array object.

Hint: The following control-flow chart might be of help:



2 Grids

a. The following class defines a two-dimensional array or a grid object with a fixed size of elements. However, executing each of the following commands either raises an error or generates an incorrect output. Extend the `iGrid` class by defining the proper function(s) to resolve those raised errors.

```
class iGrid():
    '''A representation of a two-dimensional array'''
    def __init__(self, rows, columns, fillValue = None):
        self._data = iArray(rows)
        for row in range(rows):
            self._data[row] = iArray(columns, fillValue)
```

i.
```
g = iGrid(3, 4)
print(g.getRows())
'''AttributeError: \
iGrid instance has no attribute 'getRows'''
```

ii.
```
g = iGrid(3, 4)
print(g.getColumns())
'''AttributeError: \
iGrid instance has no attribute 'getColumns' '''
```

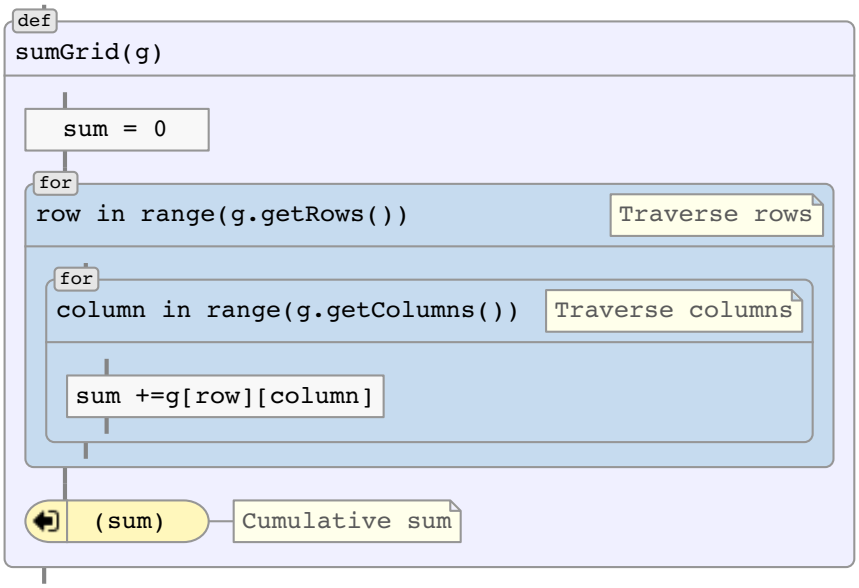
iii.
```
g = iGrid(3, 4)
print(g[0][1])
'''AttributeError: \
iGrid instance has no attribute '__getitem__''''
```

iv.
```
g = iGrid(3, 4)
print(str(g).split())
'''Output: ['<__main__.iGrid', 'instance', 'at',\
'0x10e3d9870>']''''
```

b. Describe the contents of the generated grid after running the following code.

```
iMatrix = iGrid(5,5)
for r in range(iMatrix.getRows()):
    for c in range(iMatrix.getColumns()):
        iMatrix[r][c] = r * c
```

c. Define then test a function that returns the cumulative sum of all the elements inside a given grid. Hint: The following control-flow chart might be of help.



This is the end of lab 8
Tamer Abdou, PhD